

Requerimientos en ambientes ágiles - User Stories

jueves, 17 de agosto de 2023 10:39

Procesos →

- **Procesos Definidos:**
- **Proceso Empíricos:**

Se basan en 3 pilares:

- a. Inspección
- b. Adaptación
- c. Transparencia
 - i. Hace que funcione todo lo anterior
 - ii. Marca la Claridad del Objetivo del Proyecto, en todo Momento.

Se basan en ciclos Iterativos cortos e incremental: "Así se gana experiencia "

Decide los procesos que va a utilizar, y busca lo menos que el equipo necesite utilizar.

La mejor comunicación es la

Asume procesos complicados con variables cambiantes, si el proceso se repite, los resultados obtenidos pueden ser diferentes.

- Estos procesos se ajustan de mejor forma a procesos creativos y complejos.
- La administración y el control es por medio de inspecciones frecuentes y adaptaciones.

Las personas tienen un rol fundamental en los proyectos.



Valoramos lo que está arriba mas fuerte de lo que esta por debajo.

Valora más el Vínculo de las Personas mas que el proyecto.

Valora al Cliente



El Agilismo funciona con Gente que toma decisiones.

Porcentaje de requerimientos emergentes, aparecen mientras el producto va surgiendo.

¿Qué es Ágil?

NO es una metodología o proceso

Ágil es una ideología con un conjunto definido de principios que guían el desarrollo del producto

Valores de los equipos ágiles ...

- Planificación continua, multi-nivel
- Facultados, auto-organizados, equipos completos
- Entregas frecuentes, iterativas y priorizadas
- Prácticas de ingeniería disciplinadas
- Integración continua
- Testing Concurrente

Desarrollo ágil de Software

Un compromiso útil entre nada de proceso y demasiado proceso

¿Pero qué significa Ágil?

Balance entre ningún proceso y demasiado proceso. La diferencia inmediata es la exigencia de una menor cantidad de documentación, sin embargo no es eso lo más importante:

- Los métodos ágiles son adaptables en lugar de predictivos.
- Los métodos ágiles son orientados a la gente en lugar de orientados al proceso.

Algunos frameworks ágiles

Frameworks: Un framework (o marco de trabajo) es un conjunto de herramientas, bibliotecas y reglas que proporciona una estructura prediseñada para desarrollar aplicaciones de software. Los frameworks facilitan el proceso de desarrollo al ofrecer componentes reutilizables y patrones predefinidos, lo que permite a los desarrolladores centrarse en la lógica de la aplicación en lugar de reinventar la rueda.

- FDD
- ATDD
- Scrum
- XP
- Crystal

Requerimientos en Agile

- El foco de la gestión Ágil es crear "Valor de Negocio"
- Este enfoque nos dice que tenemos que empezar con lo mínimo necesario (solo lo suficiente) para empezar a girar la rueda. Después se completa
- La parte del Desarrollo de requerimientos está enfocada a la idea de construir junto con el cliente. Como el Product Owner, que es el referente, del lado del negocio, que tiene conocimiento del producto y que trabaja junto con el equipo.

Una de las razones de cambiar el enfoque, "Solo lo suficiente":

La realidad nos muestra otra cosa



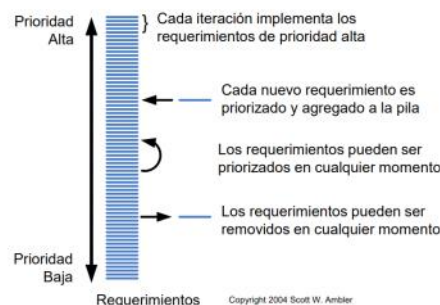
- El cliente solo llega a utilizar el 7% de lo que construimos y ponele que a veces ese 13%
- De lo que construimos Aferrarse a ese 20 %
- La gestión ágil trata de contrarrestar esto:

Esto lo Hace a través del dueño del Producto (Product Owner)

- Tiene clara cuáles son sus necesidades.
- Responsabilidad principal es Priorizar requerimientos (Decidir que realmente necesita)
- Lo ubica dentro de un contenedor (Product Backlog) por prioridad
- Si el Product Owner no participa, nosotros tomamos la decisión de Priorizar
- Si el dueño del producto tiene las ganas de estar Disponible para el equipo, no hace falta tanta documentación formal, ya que se compensa con la comunicación cara a cara

Gestión Ágil de Requerimientos de Software

Los requisitos cambiantes son una ventaja competitiva si puede actuar sobre ellos



Just in time (Analice cuando lo necesite, no antes, ni después):

Desperdicio: Especificar requerimientos que después cambian

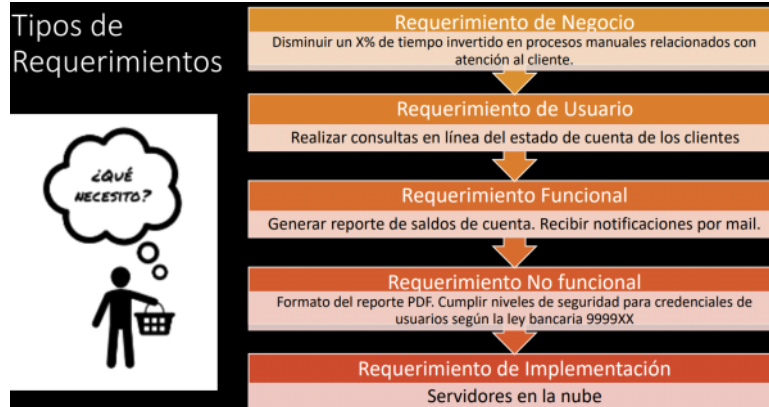
Lo que nos Dice Just in time, es que el producto no va a estar especificado al 100 al principio, ya que vamos a ir encontrando y describiendo requerimientos conforme nos haga falta

El cara-a-cara permite que fluya información vocal, subvocal, gestual con realimentación rápida.



- El contenido aporta en la comunicación solo el 17%
- El otro 83%, es como decirlo.

Tipos de Requerimientos



- Las User Story son requerimientos de Usuario, Alineado con un requerimientos de negocio



Triangulo de hierro

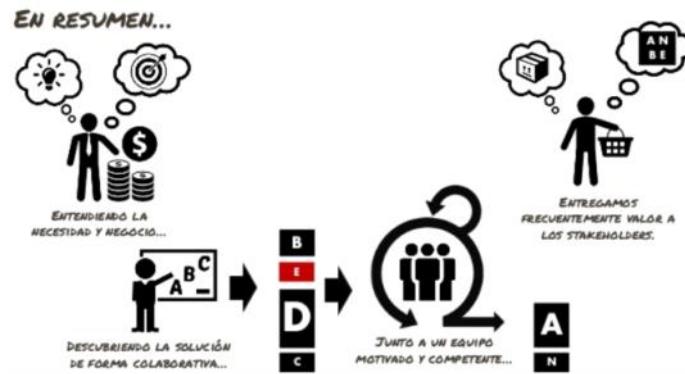


Son las Dimensiones que tenemos que tomar a la hora de trabajar.

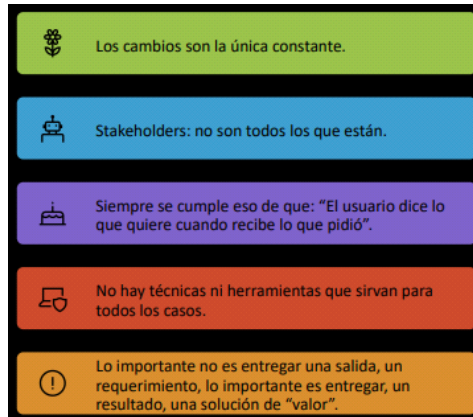
- Fijan los requisitos al Principio, y de ahí determinan los recursos que hacen falta. Y el tiempo de trabajo.
- Si cambian los requisitos, cambian los recursos. Esto no funciona

Aplicando Ágil

- Vamos a dejar fijo el TIEMPO con iteraciones de duración fija (sprint) y vamos a dejar RECURSOS fijos (equipo de trabajo con toda la infraestructura)
- Con eso fijo, cuando del producto puedo construir ?. Con esto se acuerda el ALCANCE. Cuanto de software funcionando se va a entregar.
- El núcleo de nuestro proceso es el de entregarle valor al cliente por medio de entregables funcionales en cortos periodos de tiempo.



Que Asumimos ?



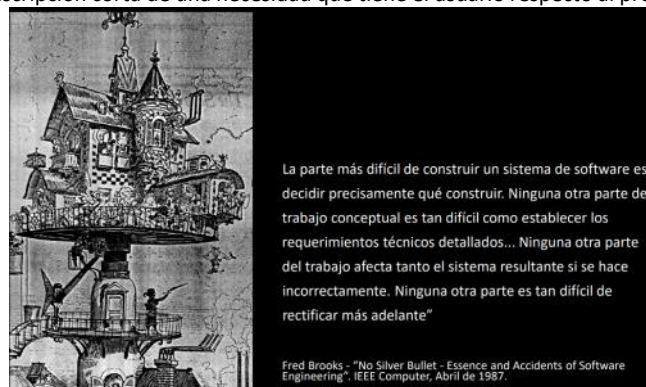
Principios Ágiles relacionados a los Requerimientos Ágiles



User Stories

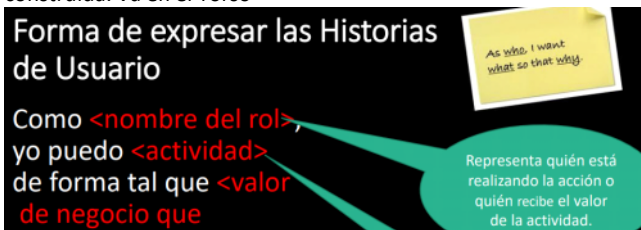
Una historia de usuario

Definición: Es una descripción corta de una necesidad que tiene el usuario respecto al producto de Software



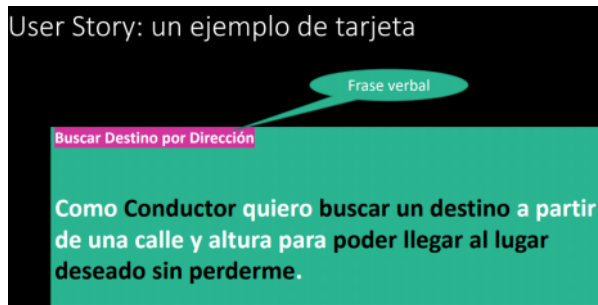
¿Cuáles son las partes de una User Story?

- Conversación:** Esto es lo más Importante y no queda en ningún lado.
- Tarjeta:** Parte que queda en algún lugar. Va en el Frente
- Confirmación:** Son las pruebas de Usuario que se identifican necesarias para hacerle después a la funcionalidad una vez que este construida. Va en el Torsó





La más importante es la Valor del Negocio, que está en el "Why". Esto para que el Produce Owner Priorice el Product Backlog



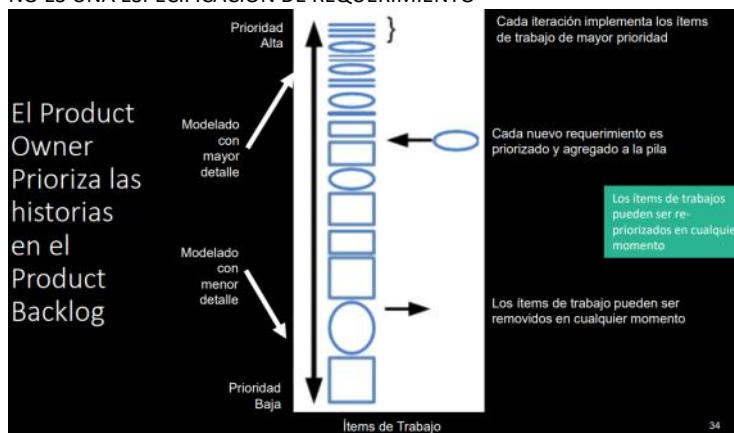
- Frase Verbal: Es una forma corta para referenciar la user Story

Las User Stories

Es producto, son partecitas de lo que tengo que hacer

Las User Stories son Multipropósito

- Las historias son:
 - Una necesidad del usuario
 - Una descripción del producto
 - Un ítem de planificación: Gestion de Proyecto
 - Token para una conversación:
 - Mecanismo para diferir una conversación
 - NO ES UNA ESPECIFICACION DE REQUERIMIENTO



Las User Stories, son porciones Verticales.

- Deben entregarse de forma que sean consumidas, osea que le sirva. Por esto se corta un poco de cada uno.
- Debe entregarse valor



MODELO DE ROLES

Modelado de Roles: Tarjeta de Rol de Usuario

Rol de Usuario: Reclutador Interno

Nos es un experto en computadoras, pero bastante adepto a utilizar la Web. Utilizará el software con poca frecuencia pero muy intensamente. Leerá anuncios de otras compañías para averiguar cuál es la mejor palabra para sus anuncios. La facilidad de uso es importante, pero más importante es que lo que aprenda, lo pueda recordar meses después.

¡SERVO A VOZ!
¿Qué rol es el reclutador interno?
¿Qué rol es el reclutador interno?
¿Qué rol es el reclutador interno?
¿Qué rol es el reclutador interno?
¿Qué rol es el reclutador interno?

- Se describe un perfil de usuario para entenderlo y para quien diseñe la experiencia de usuario, lo haga pensando en ese perfil.

Modelado de Roles: Técnicas Adicionales

• Personas

Mario trabaja como reclutador en el departamento de Speedy Networks, una fábrica de componentes de red de alta gama. El ha trabajado para Speedy Networks por 6 años. Mario tiene un arreglo de horario flexible y trabaja desde casa cada viernes. Mario es muy fuerte con las computadoras y se considera a si mismo un usuario avanzado de los productos que usa. La esposa de Mario, Kim, está terminando su Doctorado en Química en la Universidad de Stanford. Dado que Speedy Networks ha estado creciendo consistentemente, Mario siempre está buscando ingenieros.



Modelado de Roles: Técnicas Adicionales – Personajes Extremos

Diseño de un PDA para:

- El Papa
- Una mujer de 20 años con muchos novios
- Un traficante de drogas

Tanto la mujer como el traficante desearán mantener agendas separadas en caso de que la vea la policía o un novio. El Papa probablemente tenga menos necesidad de discreción pero querrá un tamaño de fuente más grande.

Usuarios Representantes (Proxies) / EVITARLOS

Que pasa cuando la situación ideal de Product Owner no es Disponible

- Tipos de usuarios representantes (Usuarios del Negocio):
 - Gerentes de Usuarios
 - Gerentes de Desarrollo
 - Alguien del grupo de marketing
 - Vendedores
 - Expertos del Dominio
 - Clientes
 - Capacitadores y personal de soporte.

Criterios de Aceptación de Historias de Usuario

Hablando mas sobre la tercera C de las User Story (CONFIRMACION)

Para poder llegar a escribir esas pruebas de Aceptación, hay que construir Criterios de Aceptación

Criterio de Aceptación: Es información concreta que tiene que servirnos para saber si lo que implementamos es correcto o no.

- Definen límites para una user story (US)
- Ayudan a que los PO respondan lo que necesitan para que la US provea valor (requerimientos funcionales mínimos)
- Ayudan a que el equipo tenga una visión compartida de la US
- Ayudan a desarrolladores y testers a derivar las pruebas.
- Ayudan a los desarrolladores a saber cuándo parar de agregar funcionalidad en una US

User Story: un ejemplo de tarjeta

Buscar Destino por Dirección

Como Conductor quiero buscar un destino a partir de una calle y altura para llegar al lugar deseado sin perderme.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

¿Cuáles son los Criterios de Aceptación buenos?

- Definen una intención, no una solución
 - Ej.: El usuario debe elegir al menos una cuenta para operar
- Son independientes de la implementación
- Relativamente de alto nivel, no es necesario que se escriba cada detalle

¿Y los detalles? ¿Dónde van?

- Detalles como:
 - El encabezado de la columna se nombra "Saldo"
 - El formato del saldo es 999.999.999,99
 - Debería usarse una lista desplegable en lugar de un Check box.
- Estos detalles que son el resultado de las conversaciones con el PO y el equipo puede capturarlos en dos lugares:
 - Documentación interna de los equipos
 - Pruebas de aceptación automatizadas

Pruebas de Aceptación de Historias de Usuario

Expresan detalles resultantes de la conversación

Complementan la User Story

Proceso de dos pasos:

1. Identificarlas al dorso de la US.
2. Diseñar las pruebas completas

User Story: Tarjeta y Pruebas de Aceptación

Buscar Destino por Dirección

Como Conductor quiero buscar un destino a partir de una calle y altura para poder llegar al lugar deseado sin perderme.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

Pruebas de Usuario

- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa).
- Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla).
- Probar buscar un destino en un país inexistente (falla).
- Probar buscar un destino en País existente, ciudad inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y demora más de 30 segundos (falla).

- Las pruebas de aceptación están relacionado con los criterios de aceptación
- Se contemplan pruebas de éxito y de fracaso

Ejemplo: para un software de un GPS

Como Conductor quiero buscar un destino a partir de sus coordenadas para conocer el camino a recorrer para llegar al destino deseado.

Ejemplo:

Como Conductor quiero buscar un destino a partir de sus coordenadas para conocer el camino a recorrer para llegar al destino deseado.

Criterios de Aceptación: Las coordenadas se representan con tres números que indican longitud y tres números que indican latitud. Cada número representa los grados, minutos y segundos respectivamente. Además se debe indicar la orientación (norte, sur, este, oeste).

- Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes (pasa).
- Probar buscar un destino en un país y ciudad existentes, de una coordenada inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes sin indicar la orientación (falla).
- Probar ingresar coordenadas de latitud y longitud válidas (pasa).
- Probar ingresar coordenadas de latitud y longitud inválidas (falla).

Definición de listo – Definition of Ready:

Es una medida de calidad que construye el equipo para poder determinar que la User esta en condiciones de entrar a una iteración de desarrollo.

Formas de controlar la Calidad que construye el equipo

Y Cuando esta Lista ? Cuando cumpla con la Definición de Hecho

Definición de Hecho – Definition of Done

- El equipo la construye en forma conjunta

Es un checkList. Que Debe cumplir lo Siguiente:

*** INVEST Model**

- **Independent** – calendarizables e implementables en cualquier orden
- **Negotiable** – el “qué” no el “cómo”
- **Valuable** – debe tener valor para el cliente
- **Estimable** – para ayudar al cliente a armar un ranking basado en costos
- **Small** – deben ser “consumidas” en una iteración
- **Testable** – demostrar que fueron implementadas

* “INVEST in Stories” – Bill Wake
<http://xp123.com/xplor/xp0308/index.shtml>

GOOD USER STORIES

I NDEPENDENT
N EGOTIABLE
V ALUABLE
E STIMABLE
S MALL (SIZED APPROPRIATELY)
T ESTABLE

Independiente: Yo le doy la libertad al Product Owner de que las priorice para que se puedan desarrollar en cualquier orden

Negociable: La User debe estar escrita en términos de “que” necesita el Usuario, no “como” lo vamos a Implementar

Valuable: Debe tener valor para el Cliente, El “Para que”

Estimable: Es asignarle un peso a la User

Small: Debe ser “consumida” en una Iteración, esto depende de la experiencia del equipo.

Testable: Pruebas de aceptación, que es demostrar que esa User efectivamente se implementó cumpliendo los criterios de Aceptación que se definieron

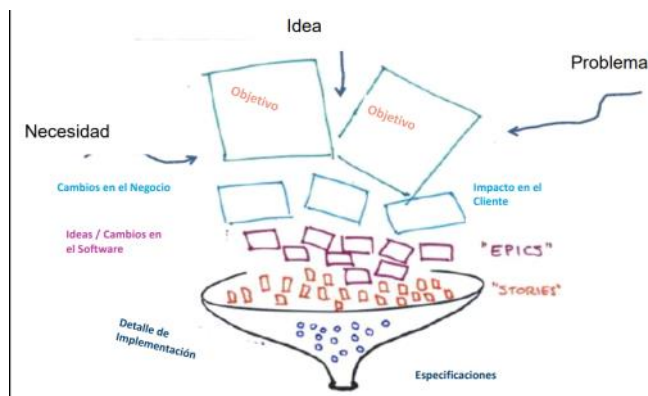
Este modelo es de mínima, el modelo que se usa para construir el Definition of Ready. Pero puede pedir que cumpla más cosas.

Algo más sobre las User Stories...

- No son especificaciones detalladas de requerimientos (como los casos de uso)
 - Son necesidades planteadas desde el punto de vista de los clientes
- Son expresiones de intención, “es necesario que haga algo como esto...”
- No están detallados al principio del proyecto, elaborados evitando especificaciones anticipadas, demoras en el desarrollo, inventario de requerimientos y una definición limitada de la solución.
- Necesita poco o nulo mantenimiento y puede descartarse después de la implementación.
- Junto con el código, sirven de entrada a la documentación que se desarrolla incrementalmente después
- No son Una ERS



- **Epic:**
 - Esta es una Historia de Usuario Grande. Como definimos que es una historia grande ?
 - Cuando no se cumple que no es INVEST
 - Cuando no lo podemos ejecutar en un Sprint
 - Se dividen es USER
- **Thema:**
 - Los Themas son un agrupador de User Story de módulos.
 - Los módulos son como subsistemas



Quando entra en el Embudo, significa que ya empezó la Iteración. Llamado Sprint Backlog en Scrum.
La imagen muestra un Nivel Alto de Abstracción (ideas, problemas) y como se va refinando hasta llegar al embudo
Esta imagen representa muy bien cómo se trabaja los requerimientos en Agile

Spikes

Son un tipo especial de User Story

Se las utiliza para disminuir incertidumbre de una User Story, o investigar algo que nos haga disminuir el riesgo

Se planifican como historias normales

Se clasifican en Técnicas y Funcionales

Pueden utilizarse para:

- Inversión básica para familiarizar al equipo con una nueva tecnología o dominio.
 - Esto si el equipo está encarando alguna tecnología que no conoce, esto hace que necesitemos un Spike o mas de uno para investigar las características de esa tecnología
- Analizar un comportamiento de una historia compleja y poder así dividirla en piezas manejables.
 - Ejemplo: Si necesitamos hacer un módulo de facturación. El negocio tiene claro que características tiene que cumplir ese módulo de facturación ?
 - Si es que no, se pueden ir dividiendo las Historias, pero también pongamos un Spike para investigar / refinar realmente lo que necesitamos.
- Ganar confianza frente a riesgos tecnológicos, investigando o prototipando para ganar confianza.
 - A veces se tiene que hacer prototipos para poner a prueba diferentes criterios que se pida
- Frente a riesgos funcionales, donde no está claro como el sistema debe resolverla interacción con el usuario para alcanzar el beneficio esperado.
 - La interacción hombre, maquina hay que investigarlo.

Técnicas	Funcionales
<ul style="list-style-type: none"> • Utilizadas para investigar enfoques técnicos en el dominio de la solución. <ul style="list-style-type: none"> • Evaluar performance potencial • Decisión hacer o comprar • Evaluar la implementación de cierta tecnología. • Cualquier situación en la que el equipo necesite una comprensión más fiable antes de comprometerse a una nueva funcionalidad en un tiempo fijo. 	<ul style="list-style-type: none"> • Utilizadas cuando hay cierta incertidumbre respecto de cómo el usuario interactuará con el sistema. • Usualmente son mejor evaluadas con prototipos para obtener realimentación de los usuarios o involucrados.

- Algunas User Stories requieren de ambos tipos de spikes. Por ejemplo:
 - Como un cliente, quiero ver mi uso diario de energía en un histograma, para poder comprender rápidamente mi consumo de energía pasado, presente y proyectado.
- En este caso un equipo puede crear dos spikes:
 - Spike Técnico:
 - Investigar cuanto tiempo requiere actualizar un display de un cliente al uso actual, determinando requerimientos de comunicación, ancho de banda y si los datos se actualizan en formato push o pull.
 - Spike Funcional:
 - Crear un prototipo de histograma en el portal web y obtener la retroalimentación de algunos usuarios respecto del tamaño, el estilo de la presentación y los atributos gráficos.

Lineamiento para Spikes

Como es una User Story tiene que ser:

- Estimables
- Demostrables
- aceptables
 - Ósea nos vamos a poner a investigar algo y nos va a llevar tanto.
 - Si es muy largo, se tiene que ver si hay que descomponerlo en más de un Spike

La excepción, no la regla

- Toda historia tiene incertidumbre y riesgos.
- El objetivo del equipo es aprender a aceptar y resolver cierta incertidumbre en cada iteración.
- Los spikes deben dejarse para incógnitas más críticas y grandes.
- Utilizar spikes como última opción.

La spike conviene que se haga siempre en un iteración anterior a la que se está ejecutando, ósea a lo el resultado de ese Spike vaya ser usada

- Salvo que el spike sea pequeño y sencillo y sea probable encontrar una solución rápida en cuyo caso, spike e historia pueden incluirse en la misma iteración.

ALGUNAS COSAS PARA DEJAR EN CLARO

- Los Requerimientos Agile:
 - Deferir el análisis detallado tan tarde como sea posible (primero porque va a cambiar y segundo, si hay alta probabilidad de que cambie porque lo atacaríamos ahora), lo que es justo antes de que el trabajo comience.
- Hasta entonces se capturan requerimientos en la forma de User Stories
- Las user stories no son requerimientos de software, si no que son **Necesidades**, no necesitan ser descripciones exhaustivas de las funcionalidad del sistema

Tips para que las user stories sean útiles para el equipo

	Un paso a la vez (evitar la palabra "Y")		Usar palabras claras en los criterios de aceptación
	No olvides la parte invisible: la conversación		Las user stories se escriben desde la perspectiva del usuario
		No forzar todo para escribirlo como user stories	

COMPONENTES DE PROYECTO DE DESARROLLO DE SOFTWARE

Proceso de Desarrollo de Software



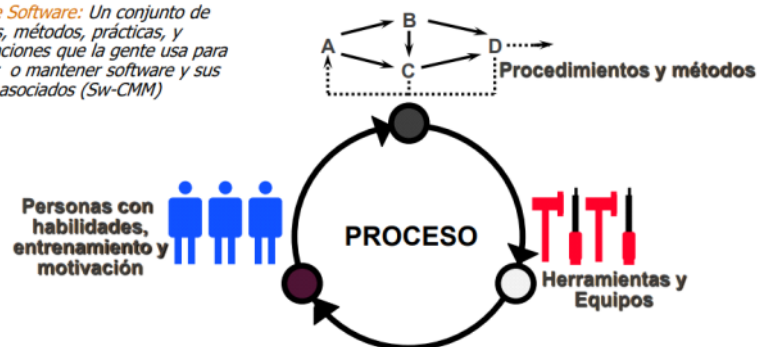
- Conjunto estructurado de actividades para desarrollar un sistema de software
- Estas actividades varían dependiendo de la organización y el tipo de sistema que debe desarrollarse.
- Debe ser explícitamente modelado si va a ser administrado.

El proceso esta formado por las 4 P

1. Procesos
 2. Personas
 3. Procedimientos
 4. Producto
5. Herramientas: son las que usamos para hacer esa transformación de que la entrada se nos transforme en la salida

Proceso: La secuencia de pasos ejecutados para un propósito dado (IEEE)

Proceso de Software: Un conjunto de actividades, métodos, prácticas, y transformaciones que la gente usa para desarrollar o mantener software y sus productos asociados (Sw-CMM)



PROCESOS EMPIRICOS

Acá sirve la experiencia, la de hoy, la del equipo, proyecto y de la gente. No sirve la experiencia Extrapolada

Empírico: Basado en la Práctica, en los hechos y no en algo que esta previamente definido.

1. Ágil
2. Lean

PROCESOS DEFINIDOS (INSPIRADOS EN LAS LINEAS DE PRODUCCION)

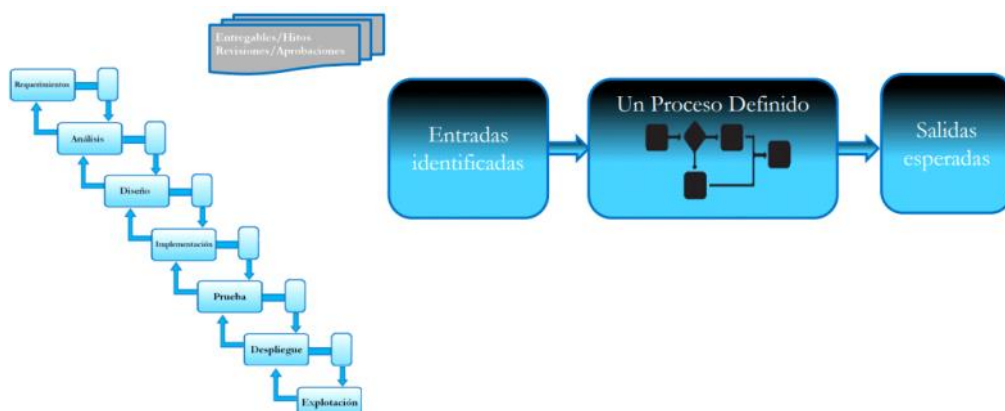
Podemos establecer, aunque determinadas entradas, si yo cumplo ciertos pasos, la salida que voy a obtener, va a ser una salida de calidad. Se puede decir que va a ser el mismo producto. Aunque la sólida del software no sea lo mismo, pero el conjunto de actividades que propone los procesos Definidos, como el Proceso Unificado de Desarrollo, son los Mismos.

Esto se construye, con el objetivo de que la salida es un producto de software de calidad

Ya tienen establecido que tiene que hacer paso a paso, entonces pueden obtener un resultado basado en esos pasos establecidos

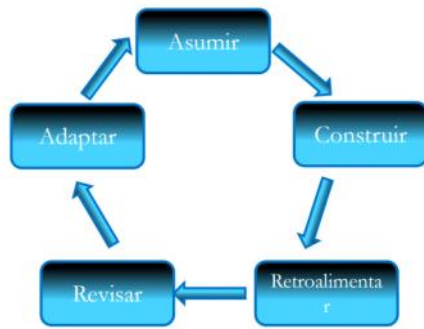
Tiene que tener Visibilidad, de donde están parados, para saber en donde terminar

Predictibilidad: Es medir, para predecir.



- Asume procesos complicados con variables cambiantes. Cuando se repite el proceso, se pueden llegar a obtener resultados diferentes.
- La administración y control se realiza través de inspecciones frecuentes y adaptaciones
- Son procesos que trabajan bien con procesos creativos y complejos.

PATRÓN DE CONOCIMIENTO EN PROCESOS EMPÍRICOS

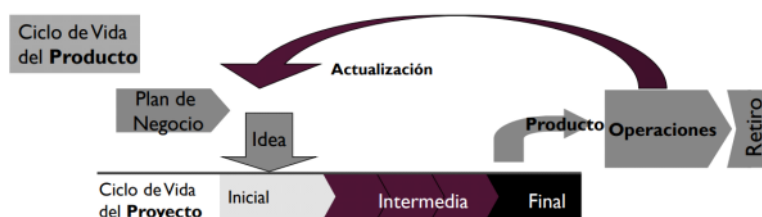


Ciclo de Mejora



Lo mismo, pero con equipos

CICLOS DE VIDA



1. Ciclo de vida del Proyecto:

Son todas las etapas por la cual pasa un proyecto.

El primer proyecto surge a partir de un plan de negocio, donde después se convierte en una idea y desde ahí se construye la primera parte del producto. Pero surgen cambios medianos que generan otro proyecto más. Y así en donde empezamos a obtener varios proyectitos hasta generar el producto

Entonces podemos decir que el ciclo de vida de un proyecto de software es una representación de un proceso. Grafica una descripción del proceso

- Los modelos especifican
 - Las fases de proceso.
 - Ejemplo: requerimientos, especificación, diseño...
 - El orden en el cual se llevan a cabo

Existen 3 ciclos de vida para un proyecto de desarrollo de software

- Secuencial: Todo a la Vez. Secuencia lineal de fases, donde cada fase debe completarse antes de pasar a la siguiente. Las fases típicas incluyen la planificación, el análisis, el diseño, la implementación, las pruebas y el mantenimiento.
 - Cascada
- Iterativo: Repetición de un conjunto de actividades en ciclos o iteraciones. En cada iteración, se llevan a cabo actividades como la planificación, el análisis, el diseño, la implementación y las pruebas, y se produce un entregable funcional al final de cada ciclo.
 - PUD
 - Alcance Fijo: Hasta que no se termina los casos de usos. Tiempo Variable
 - SCRUM:
 - Duración Fija: Se presenta tal día y no se mueve. Negociamos con el Alcance Como logramos entregar algo, cuando se nos termina el tiempo? No meter una mini cascada, porque al final no vamos a poder entregar nada Entonces Cada persona se dedica a una User, entonces es como si seria una micro cascada
- Recurso: Evalúa si seguimos estando en le proyecto, Repetición de un conjunto de actividades en ciclos o iteraciones, pero con un enfoque en la gestión de riesgos.
 - Espiral

2. Ciclo de vida del Producto

Es mas Grande que el del Proyecto

Se refiere a las diferentes etapas por las que atraviesa un producto desde su concepción hasta su retirada del mercado. Las etapas típicas del ciclo de vida del producto incluyen la introducción, el crecimiento, la madurez y el declive.

En la etapa de introducción, el producto se lanza al mercado y se llevan a cabo actividades de promoción y publicidad para darlo a conocer. En la etapa de crecimiento, el producto comienza a ganar aceptación en el mercado y las ventas aumentan rápidamente. En la etapa de madurez, las ventas se estabilizan y la competencia se intensifica. En la etapa de declive, las ventas comienzan a disminuir y el producto eventualmente se retira del mercado.



- Se adapta: El proyecto es una instanciación de ese proceso, con una representación a través de un ciclo de vida.
 - Esto se asocia a procesos Definidos.

QUE ES UN PROYECTO?

Es una unidad de gestión que nos permite obtener un producto o servicio "**único**". Básicamente en un medio de administración. Definimos un proyecto planificándolo, gestionándolo, buscando un objetivo, planificando un marco de tiempo, gestionando roles de cada uno del equipo, presupuesto y medir

Plan de Proyecto: Objetivo y Alcance del Proyecto

El objetivo del proyecto tiene relación con el objetivo del Producto:

El objetivo del proyecto es proporcionar un marco conceptual que permita al gerente hacer estimaciones razonables de recursos, costo y calendario para lograr el objetivo del producto. El objetivo del producto, por otro lado, es el resultado final que se espera lograr con el proyecto de software, como un software que cumpla con los requisitos del cliente y que sea de alta calidad

Producto: gestionar las notas de mis estudiantes

Proyecto:

Desarrollar un producto de software que gestione las notas de mis estudiantes.

El alcance sería las cosas que tengo que hacer, y lo vamos a buscar en el PROCESO: Programar,

Puede cambiar el objetivo de Proyecto siendo lo mismo el objetivo del producto y el alcance del producto no va a cambiar pero si el alcance de mi proyecto

- Un proyecto va a tener un Alcance
- Un Proyecto va a tener un Proceso y Ciclo de Vida
- Un Proyecto hace Estimaciones: Estimamos tiempo, recursos, etc. SE explica mas Abajo
- Gestión de Riesgo: Que sería la probabilidad de ocurrencia de algo que puede impactar en mi proyecto
- Métricas
- Seguimiento de Proyecto

CARACTERÍSTICAS: ORIENTACIÓN A OBJETIVOS

- Los proyectos están dirigidos a obtener resultados y ello se refleja a través de objetivos.
- Los objetivos guían al proyecto
- Los objetivos no deben ser ambiguos
- Un objetivo claro no alcanza...debe ser también alcanzable. La diferencia está que claro, no significa que sea alcanzable

Alcance del Proyecto: Todo(no te comas cosas) el trabajo y solo el trabajo que hace falta hacer para cumplir con el objetivo

CARACTERÍSTICAS: DURACIÓN LIMITADA (Fecha de Inicio y de Fin)

- Los proyectos son temporarios, cuando se alcanza el/los objetivo/s, el proyecto termina.
- Una línea de producción no es un proyecto

CARACTERÍSTICAS: TAREAS INTERRELACIONADAS BASADAS EN ESFUERZOS Y RECURSOS

- Complejidad sistémica de los problemas
- Elaboración Gradual: Te ayuda a cumplir con el Objetivo

CARACTERÍSTICAS: SON ÚNICOS

- Todos los proyectos por similares que sean tienen características que los hacen únicos.
- "No soy tan popular para ser diferente"
Homero Simpson

¿QUÉ ES LA ADMINISTRACIÓN DE PROYECTOS?

Es jugar con estos 3 conceptos, y saber como manejar la THE TRIPLE CONSTRAINT

- "...tener el trabajo hecho..." en tiempo, con el presupuesto acordado y habiendo satisfecho las especificaciones o requerimientos.
- Mas académicamente... administración de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para satisfacer los requerimientos del proyecto.
- Administrar un proyecto incluye:
 - Identificar los requerimientos

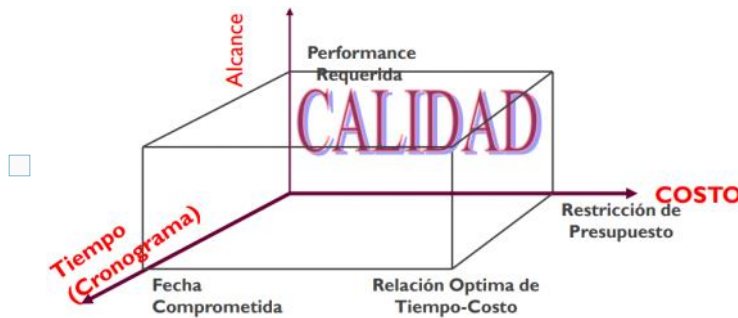
- Establecer objetivos claros y alcanzables
- Adaptar las especificaciones, planes y el enfoque a los diferentes intereses de los involucrados (stakeholders)

THE TRIPLE CONSTRAINT

- Objetivos de proyecto: que está el proyecto tratando de alcanzar?
- Tiempo: cuánto tiempo debería llevar completarlo?
- Costos: cuánto debería costar?: Esto está fundamentado con la cantidad de recursos y personas
 - El balance de estos tres factores afecta directamente la calidad del proyecto
“proyectos de alta calidad entregan el producto requerido, el servicio o resultado, satisfaciendo los objetivos en el tiempo estipulado y con el presupuesto planificado.”

Es responsabilidad del Líder de proyecto balancear estos tres objetivos (que a menudo compiten entre ellos)

Esto nos define la :



Diferencia de Proceso y ciclo de vida

La diferencia está en la abstracción, el ciclo de vida te dice en qué momento, el orden, la prioridad, etc. En cambio, el Proceso te dice el qué y el cómo.

ESTIMACIONES DE SOFTWARE.

Lo primero que estimamos es el TAMAÑO, ósea Producto, que es lo que quieres?. (EL QUE)

El software se estima en Casos de Usos, pero también se usa, aunque no sirva, líneas de código. Se cuentan Clases, requerimientos, alcances, etc.

También estima ESFUERZO, que sería el tiempo de trabajo de forma lineal, esto se mide en Horas. (EL COMO)

Otra cosa es el TIEMPO, esto se suele medir en días, ósea depende mucho del proyecto. El esfuerzo se puede transformar en Tiempo, esto viendo la gente disponible, viendo la cantidad de días de trabajo y ahí vemos el día de solapamiento, para saber cuando va a empezar cada uno. (EL CUANDO)

Y por último es el COSTO, esto se calcula al último, porque va a cambiar dependiendo, el esfuerzo, el tiempo, y el tamaño. () También se estiman los Riesgos. Es algo que puede llegar a pasar. Esto se mide por la probabilidad y el impacto, esto nos sirve para comparar los riesgos, que se lo hace de cierta forma.

Recurso Crítico: La licencia de software, los materiales, etc

Para Estimar estas cosas, hay ciertas herramientas y patrones

Unidad 1 Completa

Lo primero que hay que estudiar son las siglas, y responder la pregunta de qué es ? Y Para qué sirve ?

SOFTWARE CONFIGURATION MANAGEMENT (SCM)

La ing en software: es un concentrador de muchas disciplinas

Disciplinas Técnicas: Requerimiento, análisis, diseño, programación, prueba, Despliegue
Las técnicas se dedican al Producto con calidad

Aca tenemos puesto el Foco

Disciplinas de Gestión: Planificación y Seguimiento y supervisión:

Aca hablamos de Proyecto

Disciplinas de Soporte: SCM, Aseguramiento de Calidad de proceso y producto (PPQA) y Métricas

Son disciplinas Protectoras:

Vínculo proceso-proyecto-producto en la gestión de un proyecto de desarrollo de software.

El vínculo entre proceso, proyecto y producto es fundamental en la gestión de un proyecto de desarrollo de software. En el capítulo 24 del PDF, se explica que la administración efectiva de un proyecto de software se enfoca en las cuatro P: personal, producto, proceso y proyecto. El gerente de proyecto debe asegurarse de que estos cuatro elementos estén en equilibrio para lograr el éxito del proyecto.

El proceso se refiere a las actividades y tareas necesarias para desarrollar el software. El gerente de proyecto debe asegurarse de que el proceso sea eficiente y efectivo, y que se sigan las mejores prácticas de ingeniería de software.

El proyecto se refiere a la planificación, organización, monitoreo y control del proyecto de software. El gerente de proyecto debe asegurarse de que el proyecto se esté llevando a cabo de acuerdo con el plan y que se estén cumpliendo los plazos y presupuestos.

El producto se refiere al software que se está desarrollando. El gerente de proyecto debe asegurarse de que el producto cumpla con los requisitos del cliente y que sea de alta calidad.

En resumen, el vínculo entre proceso, proyecto y producto es fundamental en la gestión de un proyecto de desarrollo de software. El gerente de proyecto debe asegurarse de que estos tres elementos estén en equilibrio para lograr el éxito del proyecto.

Componentes de un Proyecto de Sistemas de Información.

1. Planificación del proyecto: se refiere a la definición de los objetivos del proyecto, la identificación de los recursos necesarios, la estimación de los plazos y costos, y la definición de los roles y responsabilidades del equipo de proyecto.
2. Análisis de requerimientos: se refiere a la identificación y documentación de los requerimientos del sistema de información. Esto incluye la identificación de los usuarios del sistema, la definición de las funciones y características del sistema, y la identificación de los datos necesarios para el sistema.
3. Diseño del sistema: se refiere a la definición de la arquitectura del sistema, la definición de los componentes del sistema, y la definición de las interfaces entre los componentes.
4. Implementación del sistema: se refiere a la codificación del sistema, la integración de los componentes del sistema, y la realización de pruebas para asegurar que el sistema funciona correctamente.
5. Pruebas del sistema: se refiere a la realización de pruebas para asegurar que el sistema cumple con los requerimientos del usuario y que funciona correctamente.
6. Despliegue del sistema: se refiere a la instalación del sistema en el ambiente de producción y la realización de pruebas finales para asegurar que el sistema funciona correctamente en el ambiente de producción.
7. Mantenimiento del sistema: se refiere a la realización de actividades para asegurar que el sistema siga funcionando correctamente después de su despliegue. Esto incluye la corrección de errores, la realización de mejoras y la actualización del sistema para cumplir con los nuevos requerimientos del usuario.

En resumen, los componentes de un proyecto de sistemas de información incluyen la planificación del proyecto, el análisis de requerimientos, el diseño del sistema, la implementación del sistema, las pruebas del sistema, el despliegue del sistema y el mantenimiento del sistema. Cada uno de estos componentes es esencial para el éxito del proyecto de sistemas de información.

Gestión de Configuración de Software en ambientes ágiles: ¿Es necesaria? ¿Se hace? ¿Cómo?, explique justificando sus respuestas:

Gestión de Configuración de Software en ambientes ágiles: ¿Es necesaria? ¿Se hace? ¿Cómo?, explique justificando